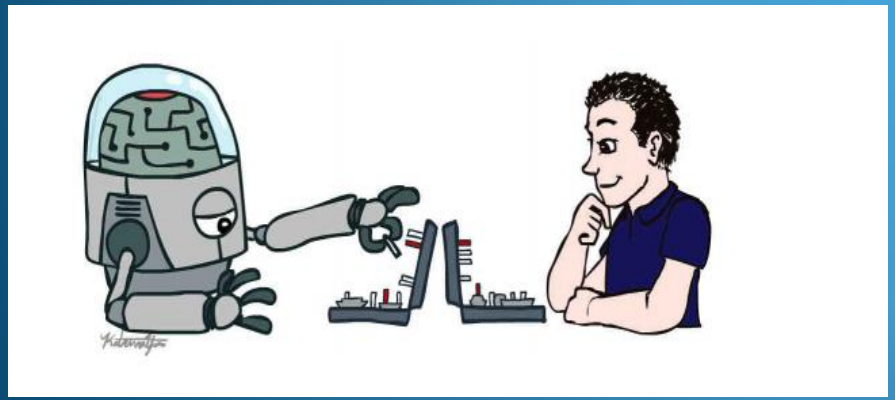
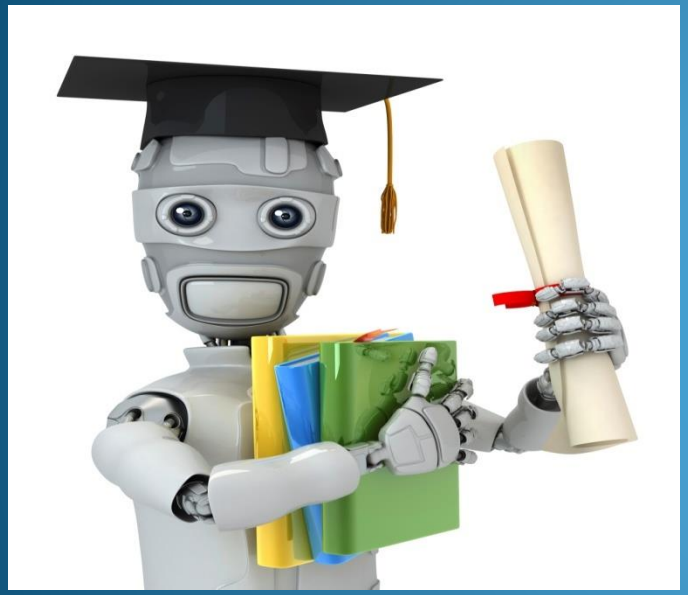
# Artificial Intelligence
# Part I: Machine learning





Andrew Ng



Machine Learning

# Logistic Regression

## Classification

## Classification

Email: Spam / Not Spam?

Tumor: Malignant / Benign ?

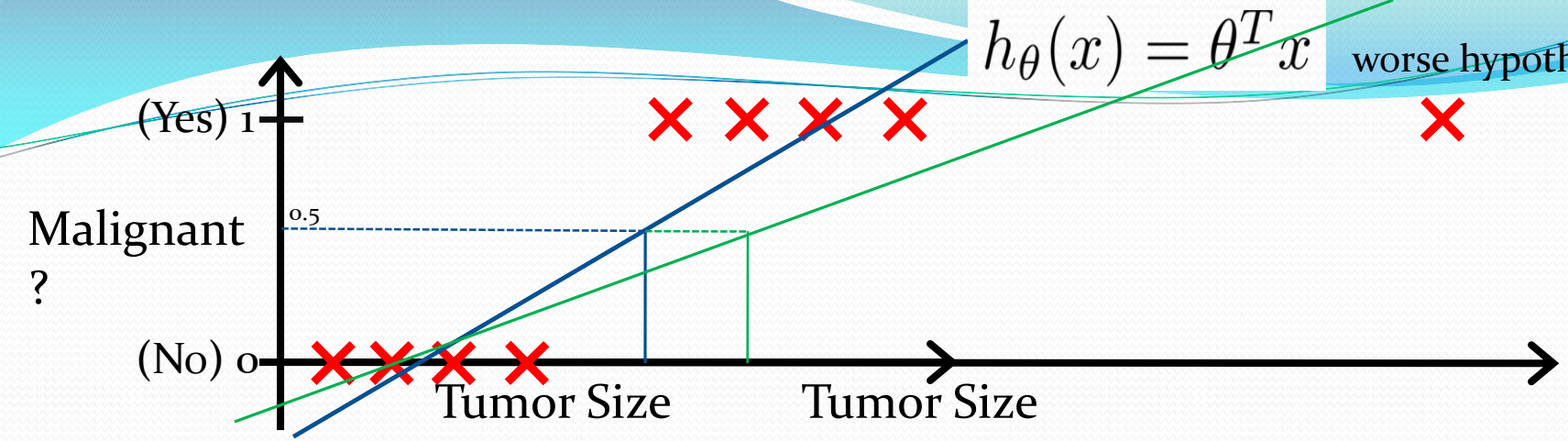
$y \in \{0, 1\}$ is binary classification problem.

0: “Negative Class” (e.g., benign tumor)

1: “Positive Class” (e.g., malignant tumor)

$y$ is the predicted variable

$y \in \{0,1,2,3\}$ is a multiclass classification problem.




Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$ , predict "y = 1"

If $h_\theta(x) < 0.5$ , predict "y = 0"

It looks like linear regression is actually doing something reasonable even though this is a classification task (good fitting to the data set)

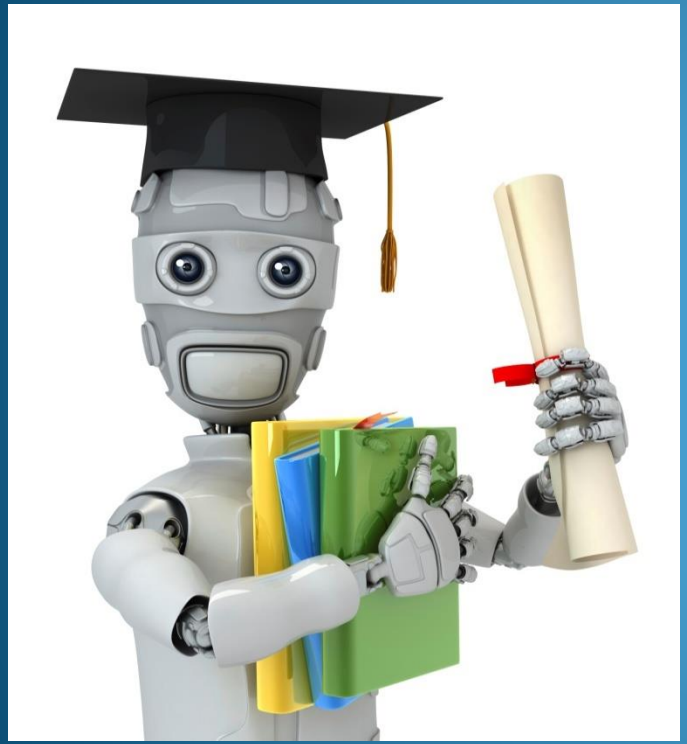
Classification: y = 0 or 1

$h_\theta(x)$ in linear regression can be > 1 or < 0

Using linear regression, the hypothesis can output values much larger than one or less than zero, even if all of good training examples have labels y equals zero or one.

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Logistic Regression = Classification



Machine Learning

# Logistic Regression

## Hypothesis Representation

## Logistic Regression Model

Want $0 \leq h_\theta(x) \leq 1$

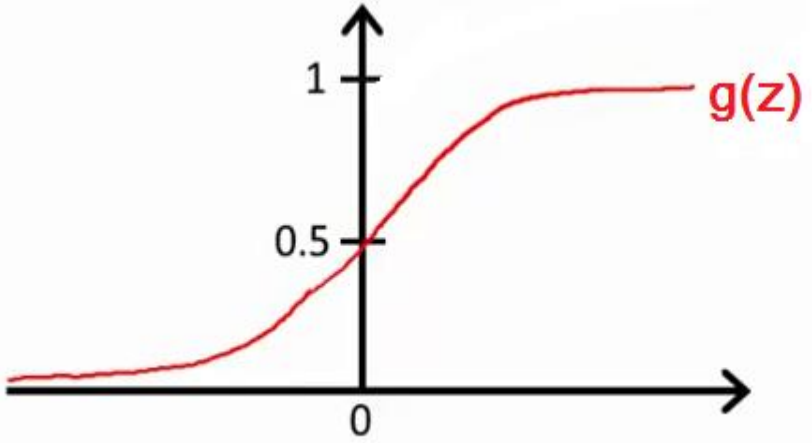
$$h_\theta(x) = g(\theta^T x)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

Sigmoid function
Logistic function

## Linear Regression Model

$$h_\theta(x) = \theta^T x$$




at z $\longrightarrow \infty$ g(z)=1
at z $\longrightarrow -\infty$ g(z)=0

## Interpretation of Hypothesis Output

$h_\theta(x)$ = estimated probability that y = 1 on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

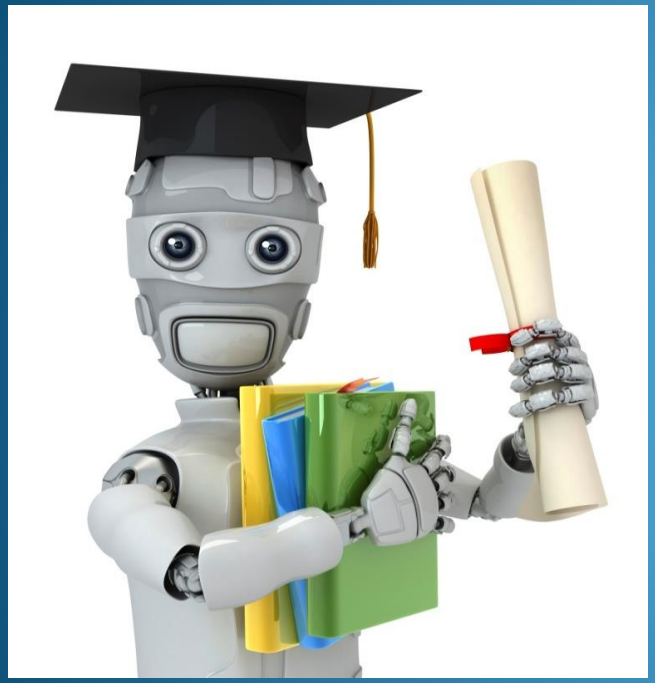
$$h_\theta(x) = 0.7$$

Tell patient that 70% chance of tumor being malignant

$h_\theta(x) = p(y=1|x;\theta)$ "probability that y = 1, given x, parameterized by $\theta$"

y=0 or 1 then

$$P(y=0|x;\theta) + P(y=1|x;\theta) = 1$$
$$P(y=0|x;\theta) = 1 - P(y=1|x;\theta)$$



Machine Learning

# Logistic Regression

## Decision boundary

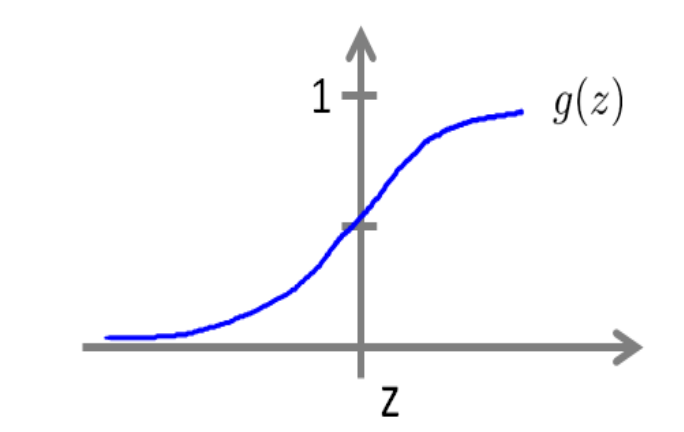
## Logistic regression

$$h_\theta(x) = g(\theta^T x) \quad g(z) = \frac{1}{1+e^{-z}}$$

$g(z) \geq 0.5 \ldots when \ldots z \geq 0 \ldots then \ldots$

$g(\theta^T x) = h_\theta(x) \geq 0.5 \ldots whenever \ldots \theta^T x \geq 0$

predict "$y = 1$" if $h_\theta(x) \geq 0.5$

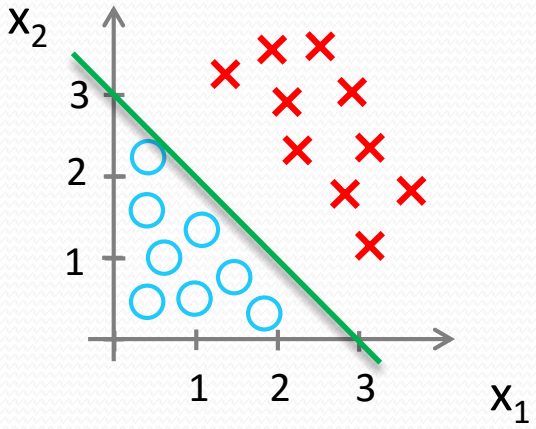



$g(z) < 0.5 \ldots when \ldots z < 0 \ldots then \ldots$

$g(\theta^T x) = h_\theta(x) < 0.5 \ldots whenever \ldots \theta^T x < 0$

predict "$y = 0$" if $h_\theta(x) < 0.5$

# Decision Boundary




$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$x_1 + x_2 = 3$ is the decision boundary where $h_\theta(x) = 0.5$

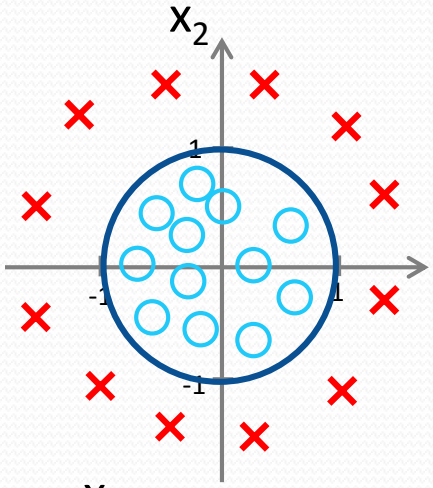
Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

$$y = 1 \ldots when \ldots g(\theta^T x) = h_\theta(x) \geq 0.5 \Rightarrow \theta^T x \geq 0$$

This means for any example of features $x_1$ and $x_2$ that satisfy this equation

$-3 + x_1 + x_2 \geq 0$ y=1

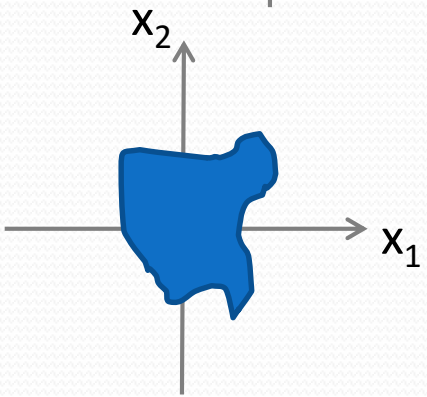
## Non-linear decision boundaries



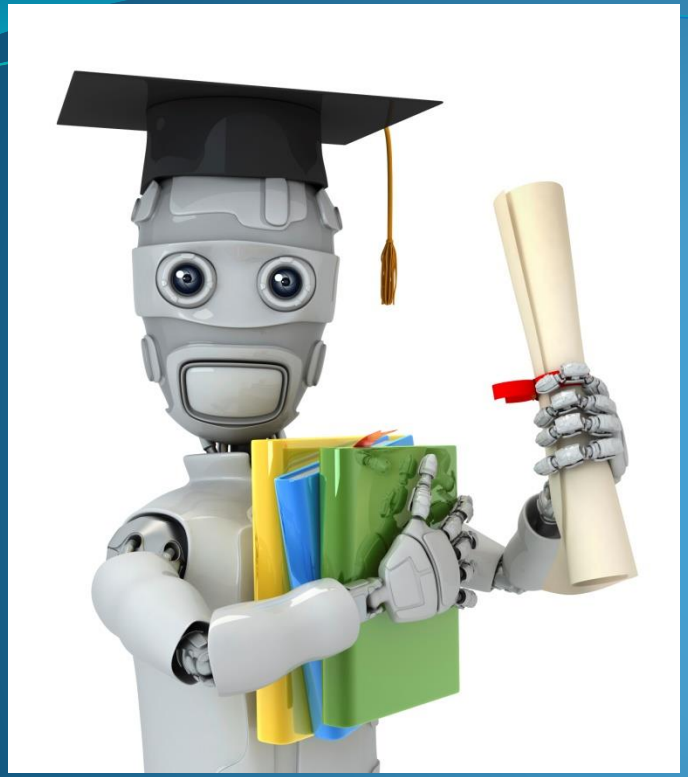

Polynomial Regression: add extra higher order polynomial features for the hypothesis

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$




$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$



Machine Learning

# Logistic Regression

## Cost function

Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

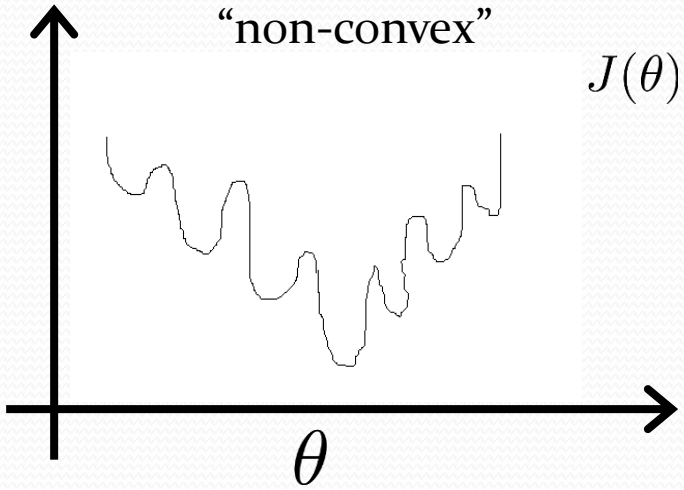
How to choose parameters $\theta$ given this training set?

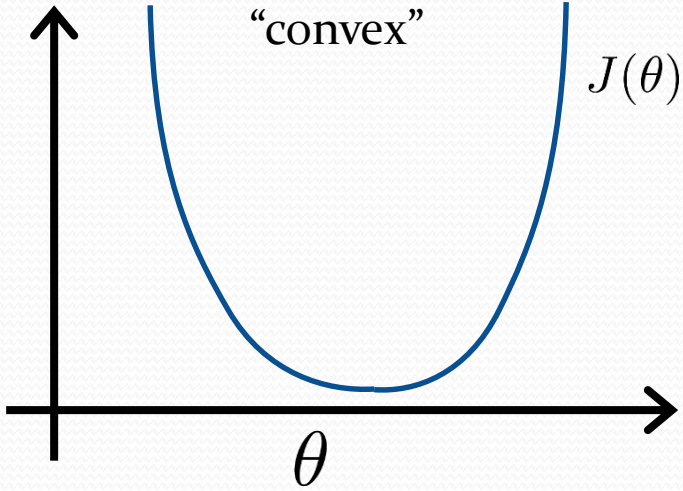
## Cost function

Linear regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Logistic regression: $\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} \left( h_\theta(x^{(i)}) - y^{(i)} \right)^2$

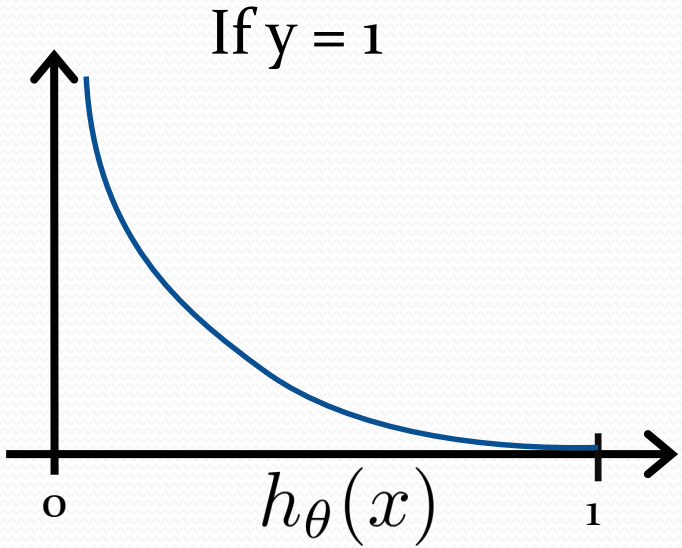
Logistic regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$



# Logistic regression cost function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



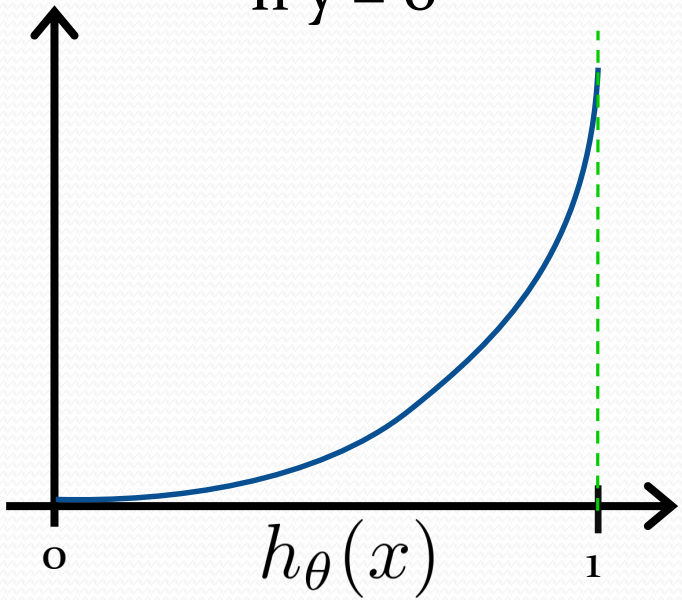

$\text{Cost} = 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

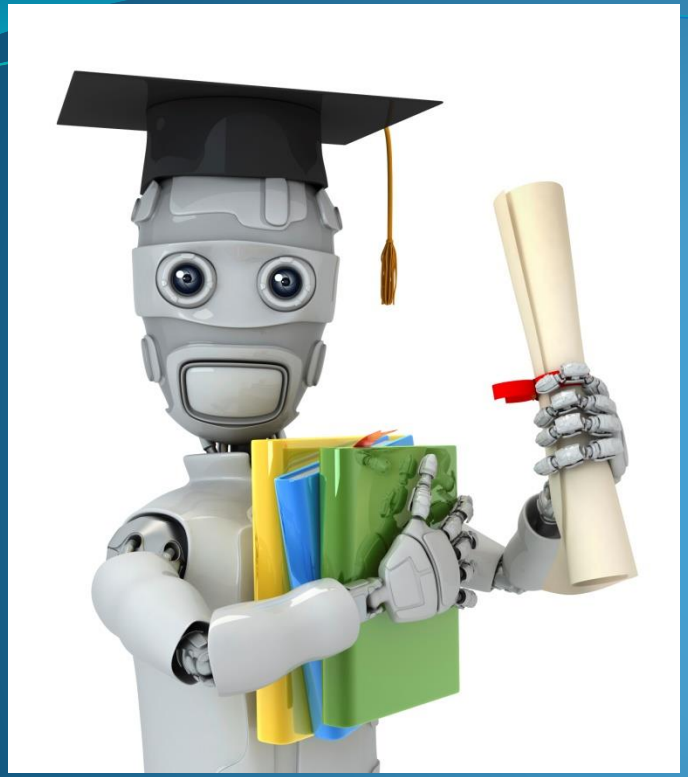
Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

# Logistic regression cost function

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

If y = 0





Machine Learning

# Logistic Regression

## Simplified cost function and gradient descent

## Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Note: $y = 0$ or $1$ always

## Simplified cost function

$Cost(h_\theta(x), y) = -y \log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$

## Logistic regression cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Cost}(h_\theta(x^{(i)}), y^{(i)})$$
$$= -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

To fit parameters $\theta$ :

$$\min_\theta J(\theta)$$

To make a prediction given new $x$:

$$\text{Output} h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

## Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$ :

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all $\theta_j$)

$\}$

## Gradient Descent

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

Want $\min_\theta J(\theta)$ :

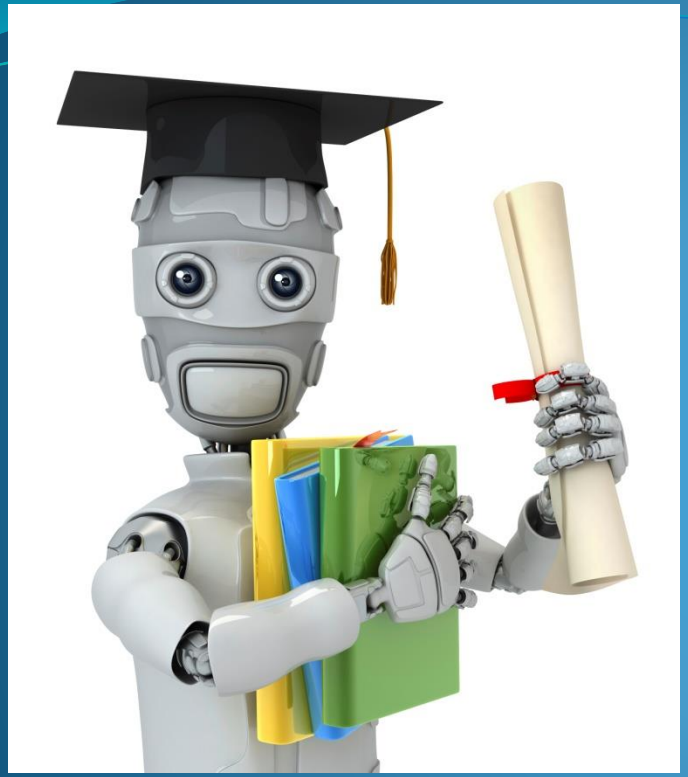
Repeat $\{$

$$\theta_j := \theta_j - \alpha \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all $\theta_j$)

$\}$

Algorithm looks identical to linear regression!



Machine Learning

# Logistic Regression

## Multi-class classification: One-vs-all

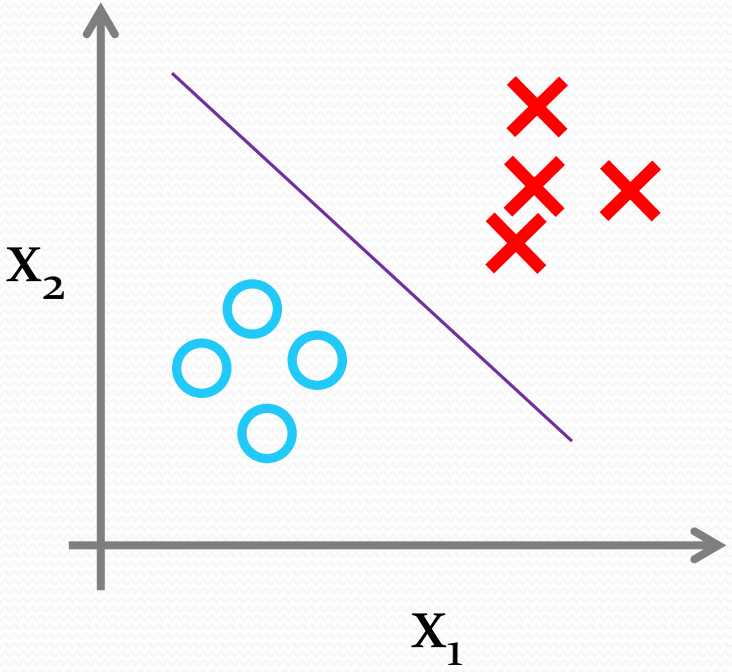
## Multiclass classification

Email foldering/tagging: Work, Friends, Family, Hobby

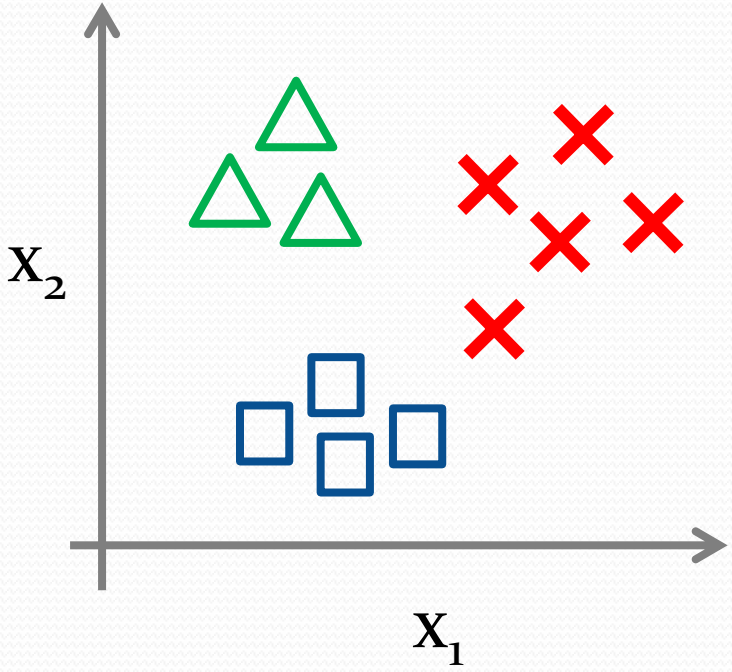
Medical diagrams: Not ill, Cold, Flu

Weather: Sunny, Cloudy, Rain, Snow

Binary classification:

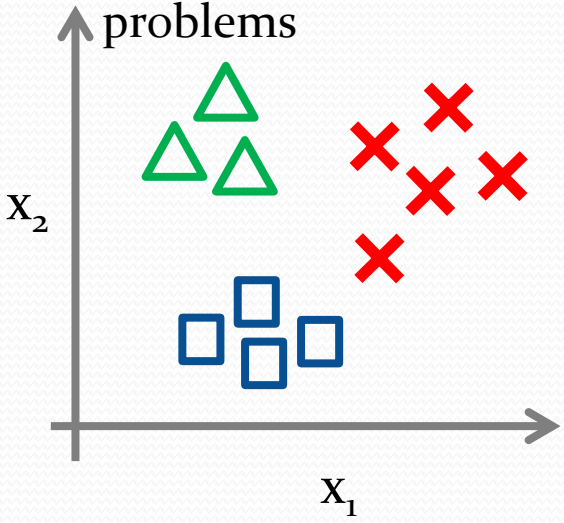



Multi-class classification:



# One-vs-all (one-vs-rest):

Turn this problem to three binary classification problems

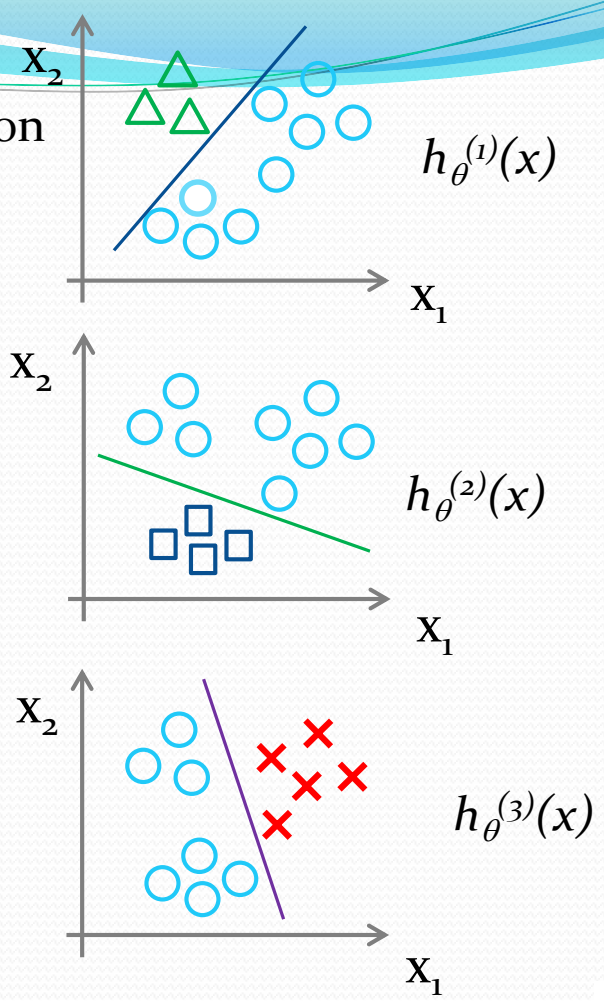



Class 1: △

Class 2: □

Class 3: ✕

$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$$

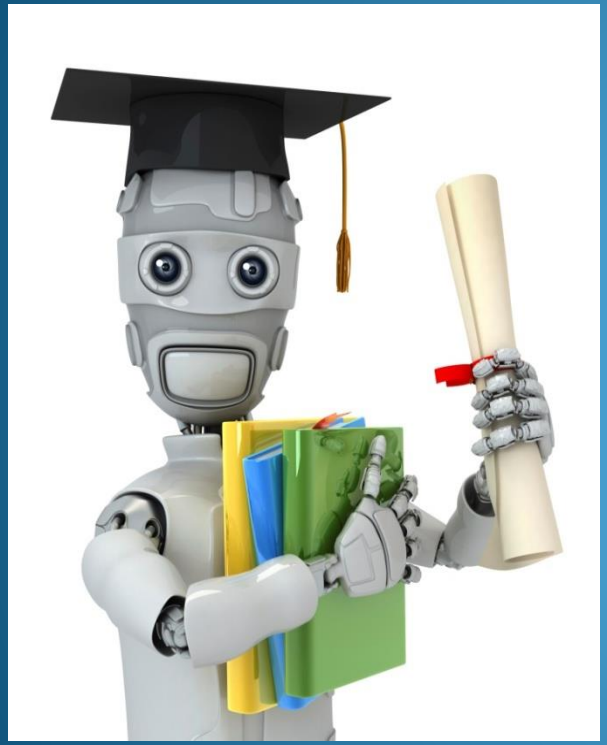


## One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class $i$ to predict the probability that $y = i$ .

On a new input $x$, to make a prediction, pick the class $i$ that maximizes

$$\max_i h_\theta^{(i)}(x)$$



# *Thanks*